Pattern Recognition Letters
journal homepage: www.elsevier.com

Identifying Intervals for Hierarchical Clustering using the Gershgorin Circle Theorem

Raghvendra Mall Mall[a,**], Siamak Mehrkanoon[a], Johan A.K. Suykens[a]

[a]*Department of Electrical Engineering, ESAT-STADIUS, Katholieke Universiteit Leuven, Kasteelpark Arenberg,10 B-3001 Leuven, Belgium*

ABSTRACT

In this paper we present a novel method for unraveling the hierarchical clusters in a given dataset using the Gershgorin circle theorem. The Gershgorin circle theorem provides upper bounds on the eigenvalues of the normalized Laplacian matrix. This can be utilized to determine the ideal range for the number of clusters (k) at different levels of hierarchy in a given dataset. The obtained intervals help to reduce the search space for identifying the ideal value of k at each level. Another advantage is that we don't need to perform the computationally expensive eigen-decomposition step to obtain the eigenvalues and eigenvectors. The intervals provided for k can be considered as input for any spectral clustering method which uses a normalized Laplacian matrix. We show the effectiveness of the method in combination with a spectral clustering method to generate hierarchical clusters for several synthetic and real world datasets.

1. Introduction

Clustering algorithms are widely used tools in fields like data mining, machine learning, graph compression, probability density estimation and many other tasks. The aim of clustering is to organize data into natural groups in a given dataset. Clusters are defined such that the data present within the group are more similar to each other in comparison to the data between clusters. Clusters are ubiquitous and application of clustering algorithms span from domains like market segmentation, biology (taxonomy of plants and animals), libraries (ordering books), WWW (clustering web log data to identify groups) and study of the universe (grouping stars based on similarity) etc. A variety of clustering algorithms exist in literature [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] etc. Spectral clustering algorithms [7, 8, 9] have become widely popular for clustering data. Spectral clustering methods can handle complex nonlinear structure more efficiently than the k-means method. A kernel-based modeling approach to spectral clustering was proposed in [10] and referred as Kernel spectral clustering (KSC). In this paper we show the effectiveness of the intervals provided by our proposed approach in combination with KSC to obtain inference about the hierarchical structure of a given dataset.

Most clustering algorithms require the end-user to provide the number of clusters (referred as k). This is also applicable for KSC. Though for KSC, we have several model selection methods like Balanced Line Fit (BLF) [10], Balanced Angular Fit (BAF)[11] and Fisher criterion to estimate the number of clusters k which are computationally expensive. However, it is not always obvious to determine the ideal value for k. It is best to choose an ideal value for k based on prior information about the data. But such information is not always available and it makes exploratory data analysis quite difficult particularly when the dimension of the input space is large.

A hierarchical kernel spectral clustering method was proposed in [14]. In order to determine the optimal number of clusters (k) at a given level of hierarchy the authors in [14] searched over a grid of values for each kernel parameter σ. They select the value of k corresponding to which the model selection criterion (BLF) is maximum. A disadvantage of this method is that for each level of hierarchy a grid search has to be performed on all the grid values for k. In [11], the authors showed that the BAF criterion has multiple peaks for different values of k corresponding to a given value of σ. These peaks correspond to optimal value of k at different levels of hierarchy. In this paper we present a novel method to determine the ideal range for k at different levels of hierarchy in a given dataset using the Gershgorin circle theorem [15].

A major advantage of the approach proposed in the paper is

**Corresponding author: Tel.: + +32 16/328657
e-mail: raghvendra.mall@esat.kuleuven.be (Raghvendra Mall Mall)

that we provide intervals for different levels of hierarchy before applying any clustering algorithm (or using any quality metric) unlike other hierarchical clustering algorithms. The Gershgorin circle theorem provides lower and upper bounds to the eigenvalues of a normalized Laplacian matrix. Using concepts similar to the eigengap, we can use these upper bounds on the eigenvalues to estimate the number of clusters at each level of hierarchy. Another advantage of this method is that we overcome the computationally expensive eigen-decomposition step. We show the efficiency of the proposed method by providing these discretized intervals (range) as input to KSC for identifying the hierarchy of clusters. These intervals can be used as starting point for any spectral clustering method which works on a normalized Laplacian matrix to identify the k clusters in the given dataset. The method works effectively for several synthetic and real-world datasets as observed from our experiments. Several approaches have been proposed to determine the ideal value of k for a given dataset [16, 17, 18, 20, 19, 21, 22, 24, 7, 8, 23, 30, 25]. Most of these methods extend the k-means or expectation maximization and proceed by splitting or merging techniques to increase or decrease the number of clusters respectively.

In this paper we propose a novel method for providing an interval (a range) for the number of clusters (k) in a given dataset. This interval helps to reduce the search space for the ideal value of k. The method uses the Gershgorin circle theorem along with upper bounds on the eigenvalues for this purpose. There are several advantages of the proposed approach. It allows us to identify intervals for the number of clusters (k) at different levels of hierarchy. We overcome the requirement of performing the eigen-decomposition step, thereby reducing the computational cost. There is no underlying assumption or prior knowledge requirement about the data.

2. Proposed Method

We consider the normalized Laplacian matrix (L) related to the Random Walk model as defined in [27]. In this model, the Laplacian matrix is defined as the transition matrix. This can mathematically be represented as $L = D^{-1}S$ where S is the affinity matrix and D is the diagonal degree matrix such that $D_{ii} = \sum_j S_{ij}$. For this model, the highest eigenvalue (equal to 1) has a multiplicity of k in case of k well-separated clusters and a gap between the eigenvalues indicates the existence of clusters. But in real world scenarios there is presence of overlap between the clusters and the eigenvalues deviate from 1. Then it becomes difficult to identify the threshold values to determine the k clusters. Therefore, we utilize the Gershgorin circle theorem to use the upper bounds on the eigenvalues to construct intervals for determining the ranges for the number of clusters (k) at each level of hierarchy in a given dataset. (If we use the normalized Laplacian [$L = I - D^{-1}S$] matrix then it would be required to use the lower bounds on the eigenvalues to construct the intervals). The actual eigenvalues are obtained by performing eigen-decomposition on Laplacian matrix L

$$Lv_j = \lambda_j v_j, j = 1, \ldots, N \tag{1}$$

where N is the number of eigenvalues.

Let $L \in \mathbb{R}^{N\times N}$ be a square matrix which can be decomposed into the sum $L = C + R$ where C is a diagonal matrix and R is a matrix whose diagonal entries are all zero. Let also $c_i = C_{ii}$, $r_{ij} = R_{ij}$ and $\bar{r}_i = \sum_{j=1}^{N} |r_{ij}|$. Then, according to the Gershgorin circle theorem [15]:

- The i^{th} Gershgorin disc associated to the i^{th} row of L is defined as the interval $I_i = [c_i - \bar{r}_i, c_i + \bar{r}_i]$. The quantities c_i and r_i are respectively referred to as the center and the radius of disc I_i respectively.
- Every eigenvalue of L lies within at least one of the Gershgorin discs I_i.
- The following condition holds:

$$c_j - \bar{r}_j \leq \bar{\lambda}_j \leq c_j + \bar{r}_j \tag{2}$$

  with $\bar{\lambda}_j$ corresponding to disc I_j. For each eigenvalue of L, λ_i, $i = 1, \ldots, N$ there exists an upper bound $\bar{\lambda}_j$, $j = 1, \ldots, N$ where i need not necessarily be equal to j. Thus, we have $\lambda_i \leq \bar{\lambda}_j$.

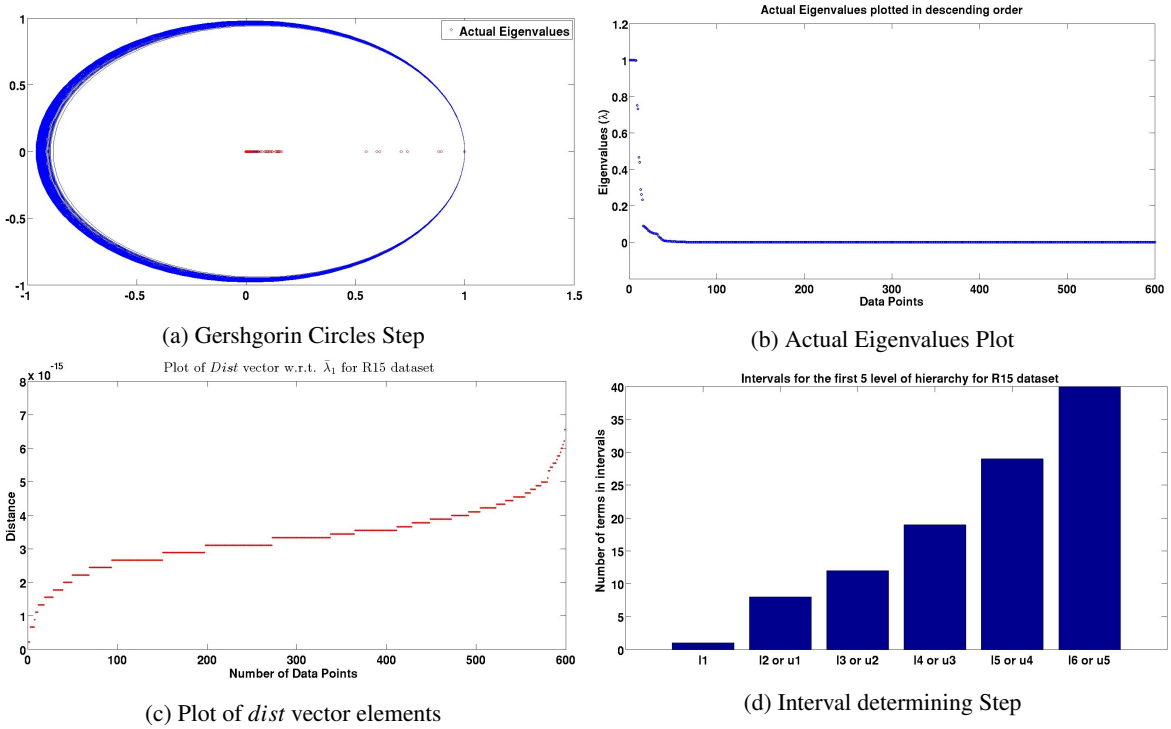
We are provided with a dataset $\mathcal{D} = \{x_1, x_2, \ldots x_N\}$ where $x_i \in \mathbb{R}^d$. We then construct the affinity matrix S by calculating similarity between each x_i and x_j. Since we use a normalized Laplacian matrix (L) the Gershgorin discs form a set of nested circles and the upper bounds i.e. $\bar{\lambda}_j = c_j + \bar{r}_j$ are all close to 1. However, these $\bar{\lambda}_j$ are more robust and the variations in their values are not as significant as the eigenvalues. It was shown in [25] that the eigenvalues are positively correlated to the degree distribution in case of real world datasets. This relation can be approximated by a linear function. We empirically observe similar correlations between the degree distribution and these upper bounds i.e. $\bar{\lambda}_j$ generated by the Gershgorin circle theorem. In [26], the authors perform stability analysis of clustering across multiple levels of hierarchy. They analyze the dynamics of the Potts model and conclude that hierarchical information for multivariate spin configuration could be inferred from spectral significance of a Markov process. In [26] it was suggested that for every stationary distribution (a level of hierarchy) the spins of the whole system reach the same value. These spin values are dependent on the different eigenvalues and the difference between the eigenvalues of the system. Inspired from this concept we propose a method to use the distance between the upper bounds to determine the intervals to search for optimal values of k for different levels of hierarchy.

We sort these $\bar{\lambda}_j$ in descending order such that $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \ldots \geq \bar{\lambda}_N$. Similarly, all the eigenvalues are sorted in descending order such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. The relation $\lambda_1 \leq \bar{\lambda}_1$ holds in accordance to the Gershgorin circle theorem. We propose a heuristic i.e. we calculate the distance of each $\bar{\lambda}_j$ from $\bar{\lambda}_1$ to obtain δ_j and maintain this value in a *dist* vector. The distance value is defined as:

$$\delta_j = Dist(\bar{\lambda}_1, \bar{\lambda}_j) \tag{3}$$

where $Dist(\cdot, \cdot)$ is the Euclidean distance function.

We then sort this *dist* vector in descending order. In order to estimate the intervals, we use a concept similar to the notion



(a) Gershgorin Circles Step

(b) Actual Eigenvalues Plot

(c) Plot of *dist* vector elements

(d) Interval determining Step

Fig. 1: Steps involved in determining the range for the number of clusters (k) at different levels of hierarchy for R15 Dataset.

of eigengap. We first try to locate the number of terms which are exactly the same as $\bar{\lambda}_1$. This can be obtained by calculating the number of terms in the *dist* vector such that $Dist(\bar{\lambda}_1, \bar{\lambda}_j) = 0$. This gives the lower limit for the first interval say $l_1 = n_1$. If there is no $\bar{\lambda}_j$ which is exactly equal to $\bar{\lambda}_1$ then the lower limit for the first interval is 1. We then move to the first term say $\bar{\lambda}_p$ in the sorted *dist* vector which is different from $\bar{\lambda}_1$. We calculate the number of terms say n_2 in the *dist* vector which are at the same distance as $\bar{\lambda}_p$ from $\bar{\lambda}_1$. The upper limit for the first interval is then defined as the sum of the lower limit and the number of terms at the same distance as $\bar{\lambda}_p$ i.e. $u_1 = n_1 + n_2$. This upper limit is also considered as the lower limit for the second interval. We continue this process till we obtain all the intervals. Since we are using the bounds on the eigenvalues ($\bar{\lambda}_j$) instead of the actual eigenvalues (λ_j), it is better to estimate intervals rather than the exact number of clusters. If the length of an interval is say 1 or 2, the search space will be too small. On the other hand, if the length of an interval is too large then we might miss hierarchical structure. So we put a heuristic that the minimum length of an interval should be 3. The intervals provide a hierarchy in a top-down fashion i.e. the number of clusters increase as the level of hierarchy increases. Algorithm 1 provides details of the steps involved to obtain the intervals for each level of hierarchy of a given dataset.

Algorithm 1: Algorithm for estimation of intervals for k

Data: Dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$
Result: Intervals for number of clusters (k) for different levels of hierarchy
Construct the affinity matrix S which comprises S_{ij}
Calculate the diagonal degree matrix $D_{ii} = \sum_{j=1}^{N} S_{ij}$.
Obtain the Laplacian matrix $L = D^{-1}S$.
Obtain the matrices C and R from L matrix using Gershgorin theorem.
Calculate $\bar{\lambda}_j = c_j + \bar{r}_j$ using C and R matrices.
Sort these $\bar{\lambda}_j$, $j = 1, \ldots, N$.
Obtain the *dist* vector by appending the distance (δ_j) of each $\bar{\lambda}_j$ from $\bar{\lambda}_1$.
Sort the *dist* vector and initialize $i = 1$ for the count of number of terms explored & $h = 1$ for the level of hierarchy.
`// Initial Condition`
Calculate $\delta_i = Dist(\bar{\lambda}_1, \bar{\lambda}_i)$.
l_h = Number of terms which have same distance as δ_i.
`// Lower limit for 1st level of hierarchy`
Increase i by l_h i.e $i := i + l_h$.
Recalculate $\delta_i = Dist(\bar{\lambda}_1, \bar{\lambda}_i)$.
$u_h = l_h$+ Number of terms which have same distance as δ_i.
`// Upper limit for 1st level of hierarchy`
while $i \leq N - 1$ **do**
  while $u_h - l_h < 3$ **do**
    Change i such that $i := u_h + 1$.
    Calculate $\delta_i = Dist(\bar{\lambda}_1, \bar{\lambda}_i)$ & $l_h = u_h$.
    Increase u_h such that $u_h = u_h$+Number of terms which have same distance as δ_i.
  end
  Increase h by 1 such that $h := h + 1$.
  $l_h = u_{h-1}$.
  Convert i to $i := u_{h-1} + 1$.
  Calculate $\delta_i = Dist(\bar{\lambda}_1, \bar{\lambda}_i)$.
  $u_h = l_h$+ Number of terms which have same distance as δ_i.
end

Figure 1 depicts the steps involved in determining the intervals for estimating the number of clusters (k) at different levels of hierarchy for the R15 [28] dataset. The R15 dataset contains 600 2-dimensional points. There are 15 clusters in this dataset. In Figure 1d, we depict the lower limit of the intervals as $l1, l2, l3, l4, l5$ and $l6$ and the upper limit of the intervals as $u1, u2, u3, u4$ and $u5$ respectively. Using these limits the first 5 intervals that we obtain for the R15 dataset are $1-8$, $8-12$, $12-19$, $19-29$ and $29-40$ respectively. These intervals are obtained using Algorithm 1. From Figure 1, we show that first we obtain the Gershgorin discs (Figures 1a) which provides us the upper bounds on the eigenvalues. This is followed by the plot

of the actual eigenvalues in descending order to show that the actual number of clusters cannot be obtained by directly using the concept of eigengap (Figures 1b) We observe from Figure 1b that the number of eigenvalues close to 1 equals 8 and the actual number of clusters in the dataset is 15. The Gershgorin discs (Figures 1a allow us to calculate the *dist* vector (Figures 1c). This enables us to determine the intervals for each level of hierarchy (Figures 1d)

In all our experiments, the affinity matrix S was constructed using the RBF-kernel. In order to handle non-linear structures, we use a kernel function to construct the affinity matrix S such that $S_{ij} = K(x_i, x_j) = \phi(x_i)^\intercal \phi(x_j)$. Here $\phi(x_i) \in \mathbb{R}^{n_h}$ and n_h can be infinite dimensional when using the RBF-kernel. One parameter of the RBF-kernel is σ. We use the mean of the multivariate rule-of-thumb proposed in [29] i.e. $\sigma = \text{mean}(\sigma(T) \times N^{-1/(d+4)})$ to estimate σ. Here $\sigma(T)$ is the standard deviation of the dataset, d is the number of dimensions in the dataset and mean is the mean value of all the $\sigma_i, i = 1, \ldots, d$.

3. Spectral Clustering

Once we obtain the intervals, we want to know the ideal value of k at each level of hierarchy. For this purpose, we provide these intervals as input to the model selection part of the Kernel Spectral Clustering (KSC) [10] method. We provide a brief description of the KSC model.

3.1. Kernel Spectral Clustering (KSC)

Given training points $\mathcal{D} = \{x_i\}_{i=1}^{N_{tr}}, x_i \in \mathbb{R}^d$. Here x_i represents the i^{th} training point and the number of points in the training set is N_{tr}. Given $\mathcal{D}$ and the number of clusters k, the primal problem of the spectral clustering via weighted kernel PCA is formulated as follows [10]:

$$\begin{aligned} \min_{w^{(l)}, e^{(l)}, b_l} \quad & \frac{1}{2}\sum_{l=1}^{k-1} w^{(l)\intercal} w^{(l)} - \frac{1}{2N}\sum_{l=1}^{k-1} \gamma_l e^{(l)\intercal} D_\Omega^{-1} e^{(l)} \\ \text{such that} \quad & e^{(l)} = \Phi w^{(l)} + b_l 1_{N_{tr}}, l = 1, \ldots, k-1 \end{aligned} \tag{4}$$

where $e^{(l)} = [e_1^{(l)}, \ldots, e_{N_{tr}}^{(l)}]^\intercal$ are the projections onto the eigenspace, $l = 1, \ldots, k-1$ indicates the number of score variables required to encode the k clusters, $D_\Omega^{-1} \in \mathbb{R}^{N_{tr} \times N_{tr}}$ is the inverse of the degree matrix associated to the kernel matrix Ω. Φ is the $N_{tr} \times n_h$ feature matrix, $\Phi = [\phi(x_1)^\intercal; \ldots; \phi(x_{N_{tr}})^\intercal]$ and $\gamma_l \in \mathbb{R}^+$ are the regularization constants. We note that $N_{tr} \ll N$ i.e. the number of points in the training set is much less than the total number of points in the dataset. Ω is obtained by calculating the similarity between each pair of points in the training set. Each element of Ω, denoted as $\Omega_{ij} = K(x_i, x_j) = \phi(x_i)^\intercal \phi(x_j)$ is obtained by using the radial basis kernel function. The clustering model is represented by:

$$e_i^{(l)} = w^{(l)\intercal} \phi(x_i) + b_l, i = 1, \ldots, N_{tr} \tag{5}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is the mapping to a high-dimensional feature space n_h, b_l are the bias terms, $l = 1, \ldots, k-1$. The projections $e_i^{(l)}$ represent the latent variables of a set of $k-1$ binary cluster indicators given by $\text{sign}(e_i^{(l)})$ which can be combined with the final groups using an encoding/decoding scheme. The dual problem corresponding to this primal formulation is:

$$D_\Omega^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)} \tag{6}$$

where M_D is the centering matrix which is defined as $M_D = \text{I}_{N_{tr}} - (\frac{(1_{N_{tr}} 1_{N_{tr}}^\intercal D_\Omega^{-1})}{1_{N_{tr}}^\intercal D_\Omega^{-1} 1_{N_{tr}}})$. The $\alpha^{(l)}$ are the dual variables and the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ plays the role of similarity function. This dual problem is closely related to the random walk model.

3.2. Hierarchical Kernel Spectral Clustering (HKSC)

The original KSC formulation [10] uses the Balanced Line Fit (BLF) criterion for model selection i.e. for selection of k and σ. This criterion works well only in case of well separated clusters. So, we use the Balanced Angular Fit (BAF) criterion proposed in [11] for cluster evaluation. It was shown in [11] that the BAF criterion has multiple peaks corresponding to different values of k for a given kernel parameter σ. In our experiments, we use the σ from the rule-of-thumb [29] as explained in Section 2. BAF is defined as:

$$\begin{aligned} &\text{BAF}(k, \sigma) = \sum_{p=1}^{k} \sum_{valid_{(i,\sigma)} \in Q_p} \frac{1}{k} \cdot \frac{MS(valid_{(i,\sigma)})}{|Q_p|} + \eta \frac{\min_l |Q_l|}{\max_m |Q_m|}, \\ &MS(valid_{(i,\sigma)}) = \max_j \cos(\theta_{j,valid_{(i,\sigma)}}), j = 1, \ldots, k \\ &cos(\theta_{j,valid_{(i,\sigma)}}) = \frac{\mu_j^\intercal e_{valid_{(i,\sigma)}}}{\|\mu_j\| \|e_{valid_{(i,\sigma)}}\|}, j = 1, \ldots, k. \end{aligned} \tag{7}$$

where $e_{valid_{(i,\sigma)}}$ represents projection of i^{th} validation point for the given σ, μ_j is mean projection of all validation points in cluster j and Q_p represents the set of validation points belonging to cluster p and $|Q_p|$ is its cardinality. BAF works on the principle of angular similarity. Validation points are allocated to the clusters to which (μ_j) they have the least angular distance. We use a regularizer η to vary the priority between angular fitting and balance. The BAF criterion varies from [-1, 1] and higher values are better for a given value of k.

So this criterion works on the intervals provided by the proposed approach to detect the ideal number of clusters (k) for each level of hierarchy in the given dataset. We then build the KSC model using that value of k and obtain the cluster memberships for all the points using the out-of-sample extensions property. In constructing the hierarchy we start with smaller values of k before moving to intervals with larger value of k. Thus, the hierarchy of clusters are obtained in a top-down fashion. One advantage of performing the KSC method is that if the actual eigenvalues are too small for a particular interval of hierarchy the KSC method will stop automatically. It suggests that KSC cannot find any more clusters for this interval and future intervals. Thus, we have reached the final level where each individual data point is a cluster.

We then use the linkage criterion introduced in [14] to determine the split of the clusters based on the evolution of the cluster memberships as the hierarchy goes down. The idea is to find the set of points belonging to different clusters at a higher level of hierarchy which are descendants of the same cluster at a lower level of hierarchy. Then, a parent-child relationship is established between these set of clusters. An important point to note is that the splits might not be perfect. For each value of k, the KSC model is run independently and nested partitions are

not always guaranteed. A cluster at higher level of hierarchy is considered as child of a cluster at lower level of hierarchy if majority of the points in this child cluster are coming from the parent cluster. A visualization of the hierarchical tree structure generated by HKSC for S1 dataset is depicted in Figure 2.

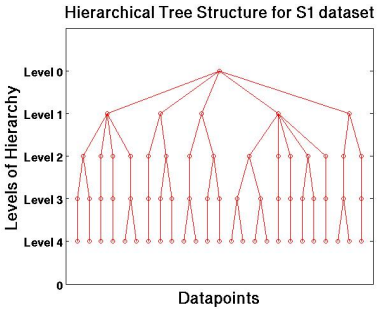


Fig. 2: Hierarchical tree structure representing the top 5 levels of hierarchy for S1 dataset using HKSC methodology.

Algorithm 2 explains the steps of hierarchical kernel spectral clustering (HKSC) algorithm that we are using in this paper.

Algorithm 2: Hierarchical Clustering Algorithm

Data: Dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ and the intervals for k provided by Gershgorin Circle Theorem.
Result: Hierarchical cluster organization for the dataset $\mathcal{D}$

1 Divide the dataset into training, validation and test set as shown in [10].
2 Use the mean of the multivariate rule-of-thumb [29] as kernel parameter σ.
3 **for** *Each Interval from Algorithm 1* **do**
4 Use the kernel parameter σ to train a KSC model using the training set.
5 Select the k from this interval corresponding to which the BAF [11] criterion is maximum and build a KSC model for k clusters.
6 Use the out-of-sample extensions property of the clustering model to obtain cluster memberships for the test set.
7 **end**
8 Stack all the cluster memberships obtained from the different intervals.
9 Create a linkage matrix as proposed in [14] by identifying which clusters split starting from the top of the hierarchy.

4. Experiments

We conducted experiments on several synthetic and real world datasets. These datasets were obtained from `http://cs.joensuu.fi/sipu/datasets/`. Table 1 provides details about these datasets along with the lower (li) and upper (ui) limit for each interval identified by our proposed method.

Table 1: Details of various datasets used for experimentation. Ideal k represents the groundtruth number of clusters available for these datasets. However, in case of real-world datasets this ideal k is not always known beforehand.

Dataset	Points	Dim	Ideal k	Level 1		Level 2		Level 3		Level 4		Level 5	
				l1	**u1**	**l2**	**u2**	**l3**	**u3**	**l4**	**u4**	**l5**	**u5**
Aggregation	788	2	7	2	5	5	15	15	21	21	26	26	31
D31	3100	2	31	1	13	13	16	16	22	22	27	27	34
DIM032	1024	32	16	1	6	6	14	14	32	32	64	64	152
DIM064	1024	64	16	1	13	13	42	42	169	169	445	445	663
DIM512	1024	512	16	1	6	6	22	22	99	99	300	300	526
DIM1024	1024	1024	16	3	35	35	188	188	426	426	641	641	768
Glass	214	9	7	1	6	6	17	17	32	32	56	56	83
Iris	150	4	3	2	6	6	15	15	35	35	49	49	83
Pathbased	300	2	3	1	6	6	13	13	22	22	37	37	58
R15	600	2	15	1	8	8	12	12	19	19	29	29	40
Sprial	312	2	3	1	17	17	30	30	49	49	85	85	137
S1	5000	2	15	1	6	6	16	16	23	23	27	27	32
Wine	178	13	3	1	5	5	10	10	22	22	34	34	59
Yeast	1484	8	10	1	10	10	15	15	21	21	27	27	38

Table 2: Hierarchical KSC (HKSC) results on various datasets used for experimentation. 'NA' here means that the eigenvalues are too small and no further clusters are detected i.e. at this level all the points are individual clusters.

Dataset	Ideal k	Level 1		Level 2		Level 3		Level 4		Level 5	
		k_1	**BAF**	k_2	**BAF**	k_3	**BAF**	k_4	**BAF**	k_5	**BAF**
Aggregation	7	3	0.934	**6**	**0.821**	16	0.695	21	0.5925	26	0.564
D31	31	4	0.829	13	0.755	19	0.837	26	0.655	**29**	**0.679**
DIM032	16	3	0.782	13	0.825	**15**	**0.841**	33	0.32	NA	NA
DIM064	16	13	0.818	**16**	**0.895**	42	0.2625	NA	NA	NA	NA
DIM512	16	3	0.721	**16**	**0.975**	22	0.5225	NA	NA	NA	NA
DIM1024	16	**16**	**0.998**	35	0.325	NA	NA	NA	NA	NA	NA
Glass	7	6	0.658	**7**	**0.677**	18	0.558	NA	NA	NA	NA
Iris	3	**3**	**0.71**	6	0.655	NA	NA	NA	NA	NA	NA
Pathbased	3	**3**	**0.888**	9	0.709	14	0.623	24	0.522	NA	NA
R15	15	7	0.844	9	0.879	**15**	**0.99**	19	0.60	NA	NA
Spiral	3	**3**	**0.818**	21	0.541	32	0.462	NA	NA	NA	NA
S1	15	5	0.842	**15**	**0.876**	16	0.805	23	0.76	NA	NA
Wine	3	**3**	**0.685**	6	0.624	10	0.5025	22	0.406	NA	NA
Yeast	10	3	0.824	**11**	**0.64**	15	0.629	26	0.651	NA	NA

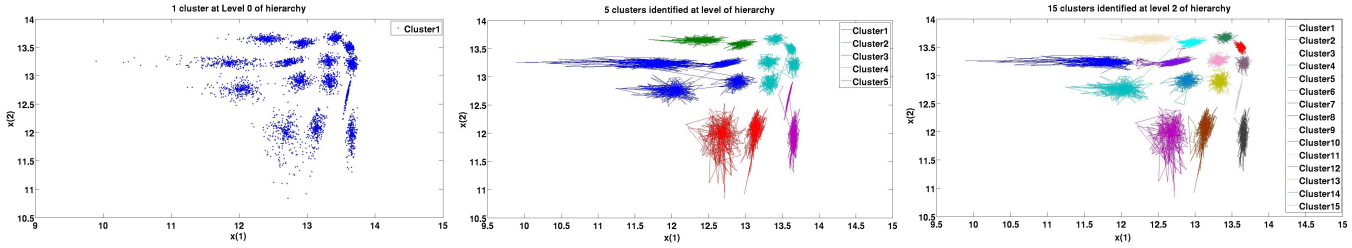
For HKSC method, we randomly select 30% of the data for training and validation respectively and the entire dataset as test set. We perform 10 randomizations of HKSC and report the mean results in Table 2. From Table 2, we observe that the HKSC method identifies the ideal number of clusters for most of the datasets including the Dim064, Dim512, Dim1024, Glass, Iris, Pathbased, R15, Spiral, S1 and Wine datasets. In most cases, the Balanced Angular Fit (BAF) values are maximum for the number of clusters identified by HKSC method which are closest to ideal number of clusters. Since the HKSC method requires to construct a kernel matrix ($N_{tr} \times N_{tr}$) in the dual, this method works best when the number of dimensions for a given dataset is large with fewer number of points.

In Figure 3 and Figure 4, we depict the clusters identified by the HKSC method for the intervals by our proposed approach at different levels of hierarchy. Figure 3 shows the results on S1 dataset whereas Figure 4 shows the results for R15 dataset. For the S1 dataset we identified 5 clusters at level 1 and 15 clusters at level 2 of hierarchy. Similarly, for the R15 dataset we identified 7 clusters at level 1, 9 clusters at level 2 and 15 clusters at level 3 of hierarchy. The clusters identified by the HKSC method for each level of hierarchy for both the datasets captures the underlying hierarchical structure. Figure 5 highlights the result of HKSC on the intervals provided by our proposed method for 2 real world images.

We compare HKSC results with linkage [31] based hierarchical clustering techniques including Single Link (SL), Complete Link (CL) and average Link (AL). The time complexity of proposed approach for identifying the intervals along with HKSC is $O(N^2 + k \times N_{tr}^3)$. But since $N_{tr} \ll N$ the overall complexity can be given as $O(N^2)$. The time complexity of SL, CL and AL are $O(N^2)$, $O(N^2 \log(N))$ and $O(N^2 \log(N))$ respectively. Since BAF criterion uses eigen-projections and is catered towards spectral clustering methods, we use another quality metric namely silhouette (SIL) [32] criterion. Higher SIL values correspond to better quality clusters. For all these methods, we compare that level of hierarchy which results in maximum SIL value as shown in Table 3.

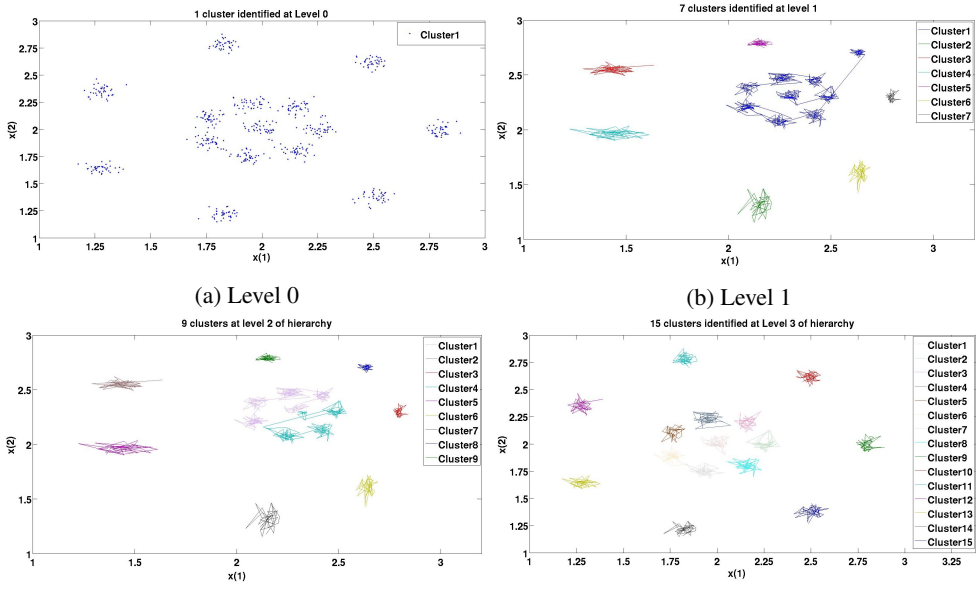
5. Conclusion

We proposed a novel method for identifying the ideal range for the number of clusters (k) at different levels of hierarchy in



(a) Level 0 (b) Level 1 (c) Level 2

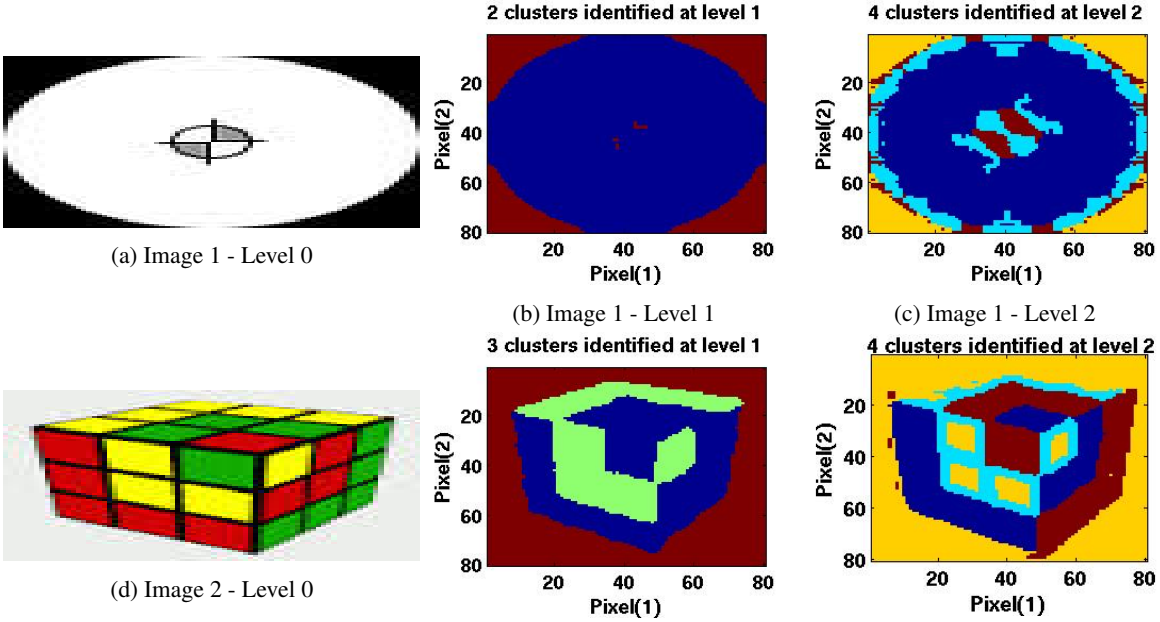
Fig. 3: Clusters identified by HKSC method at Level 1 and Level 2 of hierarchy from the intervals provided in Table 1 by proposed method for the S1 dataset.



(a) Level 0 (b) Level 1

(c) Level 2 (d) Level 3

Fig. 4: Clusters identified by HKSC method at Levels 1, 2 and 3 of hierarchy from the intervals provided in Table 1 by proposed method for the R15 dataset.



(a) Image 1 - Level 0 (b) Image 1 - Level 1 (c) Image 1 - Level 2

(d) Image 2 - Level 0 (e) Image 2 - Level 1 (f) Image 2 - Level 2

Fig. 5: Clusters identified by HKSC method at Level 1, Level 2 of hierarchy by the proposed method for the two images.

a given dataset. The proposed approach provided these intervals before applying any clustering algorithm. The proposed technique used the Gershgorin circle theorem on a normalized Laplacian matrix to obtain the upper bounds on the eigenvalues without performing the actual eigen-decomposition step. This helps to reduce the computational cost. We then obtained intervals for ideal value of k at each level of hierarchy using these bounds. We can then provide these intervals to any clustering algorithm which uses a normalized Laplacian matrix. We showed that the method works effectively in combination with HKSC for several synthetic and real world datasets.

Table 3: Comparison of various hierarchical clustering techniques. We compare that level of hierarchy corresponding to which the **SIL** quality metric is maximum. We show the number of clusters for that level of hierarchy as **Best** k. We also compare computational time (in seconds) required by the different clustering techniques. The HKSC method generally results in best quality clusters (**SIL**) along with the AL clustering technique. The HKSC and SL methods are computationally cheaper. The SL technique, though fast, results in the worst quality clusters. The best results are highlighted in bold.

Dataset	HKSC			SL			CL			AL		
	Best k	**SIL**	**Time(s)**	**Best** k	**SIL**	**Time(s)**	**Best** k	**SIL**	**Time(s)**	**Best** k	**SIL**	**Time(s)**
Aggregation	6	**0.70**	1.29	7	0.55	**1.28**	7	0.67	3.74	7	0.69	3.81
D31	29	**0.71**	22.12	30	0.64	**21.18**	30	0.68	59.56	30	**0.71**	61.12
DIM032	15	**0.86**	**2.91**	14	0.80	3.12	15	0.83	10.15	15	**0.86**	11.22
DIM064	16	**0.78**	**3.55**	16	0.64	4.23	16	0.71	12.12	16	0.74	13.87
DIM512	16	**0.68**	**5.24**	16	0.60	6.53	16	0.64	16.44	16	0.66	18.56
DIM1024	16	**0.62**	**6.72**	16	0.53	8.12	16	0.60	24.21	16	**0.62**	26.45
Glass	7	0.74	0.11	7	0.67	**0.09**	7	0.72	0.21	7	**0.75**	0.22
Iris	3	**0.95**	0.08	3	0.85	**0.05**	3	0.89	0.15	3	0.92	0.15
Pathbased	3	**0.89**	0.33	3	0.84	**0.31**	3	0.87	0.62	3	0.88	0.65
R15	15	0.78	**1.34**	15	0.74	1.35	15	0.77	2.52	15	**0.90**	2.84
Spiral	3	**0.82**	0.38	3	0.76	**0.35**	3	0.78	0.71	3	0.80	0.73
S1	15	**0.88**	**64.12**	15	0.54	65.23	15	0.79	187.9	15	0.81	191.2
Wine	3	0.65	0.10	3	0.62	**0.08**	3	0.64	0.18	3	**0.68**	0.19
Yeast	11	**0.84**	1.01	10	0.64	**0.99**	10	0.76	18.25	10	0.82	18.7

Acknowledgments

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This chapter reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants. Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. IWT: projects: SBO POM (100031); PhD/Postdoc grants. iMinds Medical Information Technologies SBO 2014. Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017.

References

[1] Steinhaus, H.: Sur la division des corp material en parties. Bulletin of Acad. Polon. Sci., vol 4(c1 3), 801–804, 1956.

[2] Llyod, S.: Least squares quantization in PCM. IEEE Transactions on Information Theory, vol 28, 129–137, 1982. Originally as an unpublished Bell Laboratories Technical Report (1957).

[3] Ester, M. , Peter, K., Hans, J., Xiaowel, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2^{nd} International Conference on Knowledge Discovery and Data Mining, AAAI Press, 226–231, 1996.

[4] Mclachlan, G.E., Basford, K.E.: Mixture Models: Inference and applications to clustering. Marcel Dekker, 1987.

[5] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of machine learning research, vol 3, 993–1022, 2003.

[6] Welling M., Rosen-zvi, M., Hinton, G.: Exponential family harmoniums with an application to information retrieval. Advances in Neural Information Processing Systems, vol 17, 1481–1488, 2005.

[7] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems, vol 14, MIT Press, 849–856, 2001.

[8] von Luxburg, U.: A tutorial on Spectral Clustering. Stat. Computat., vol 17, 395–416.

[9] Shi, J., Malik, J: Normalized cuts and image segmentations. IEEE Transactions on Pattern Analysis and Intelligence, vol 22 (8), 888–905, 2000.

[10] Alzate, C, Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 32 (2), 335–347, 2010.

[11] Mall R., Langone R., Suykens J.A.K.: Kernel Spectral Clustering for Big Data Networks. Entropy, Special Issue: Big Data, vol 15 (5), 1567–1586, 2013.

[12] Mall R., Langone R., Suykens J.A.K.: Self-Tuned Kernel Spectral Clustering for Large Scale Networks. In Proc. of the IEEE International Conference on Big Data (IEEE BigData 2013), Santa Clara, USA, Oct. 2013.

[13] Mall R., Langone R., Suykens J.A.K.: Multilevel Hierarchical Kernel Spectral Clustering for Real-Life Large Scale Complex Networks. PLOS ONE, e99966, vol 9(6), 1–18, 2014.

[14] Alzate C., Suykens J.A.K.: Hierarchical Kernel Spectral Clustering. Neural Networks, vol. 35, 21–30, 2012.

[15] Gershgorin, S.: Über die Abgrenzung der Eigenwerte einer Matrix. Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk 6, 749-754, 1931.

[16] Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of International Conference on Machine Learning, Morgan Kaufmann, 727–734, 2000.

[17] Kass, R.E., Wasserman, L.: A reference Bayesian test for nested hypothesis and its relationship to the Schwarz criterion. Journal of the American Statistical Association, vol 90(431), 928–934, 2000.

[18] Schwarz, G: Estimating the dimension of a model. The Annals of Statistics, vol 6(2), 461–464, 2001.

[19] Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control, vol 19, 716–723, 1974.

[20] Rissanen, J.: Modeling by shortest data description. Automatica, vol 14, 465–471, 1978.

[21] Hamerly, G., Elkan, C.: Learning the k in k-means. In Proceedings of 17^{th} annual conference on neural information processing systems (NIPS), 281–288, 2003.

[22] Sand, P., Moore, A.W.: Repairing faulty mixture models using density estimations. In Proceedings of 18^{th} International Conference on Machine Learning, 457–464, 2001.

[23] Polito, M., Perona,P.: Grouping and dimensionality reduction by locally linear embedding. Advances in NIPS vol. 14, 2002.

[24] Feng, Y., Hamerly, G.: PG-means: Learning the number of clusters in data. Advances in Neural Information Processing Systems 18, 2006.

[25] Chen J., Lu, J., Zhan, C., Chen, G.: Laplacian Spectra and Synchronization Processes on Complex Networks. Handbook of Optimization in Complex Networks. Optimization and its Applications, 81–113, 2012.

[26] Li, H.J., Zhang, X.S.: Analysis of stability of community structure across multiple hierarchical levels. EPL, 103 (58002), 2013.

[27] Meila, M., Shi, J.: A Random Walks View of Spectral Segmentation. In Proceedings of International Conference on Artificial Intelligence and Statistics, 2001.

[28] Veenman, C.J., Reinders, M.J.T., Backer, E.: A maximum variance cluster algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 24(9), 1273–1280, 2002.

[29] Scott, D.W., Sain, S.R.: Multi-dimensional Density Estimation. Data Mining and Computational Statistics, vol 23, 229–263, 2004.

[30] Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of Royal Statistical Society, vol 63(2), 411–423, 2001.

[31] Jain, A.K., Flynn, P.: Image segmentation using clustering. In Advances in Image Understanding, IEEE Computer Society Press, 65–83, 1996.

[32] Rabbany, R., Takaffoli, M., Fagnan, J., Zaiane, O.R., Campello, R.J.G.B.: Relative Validity Criteria for Community Mining Algorithms. IEEE/ACM ASONAM, 258–265, 2012.